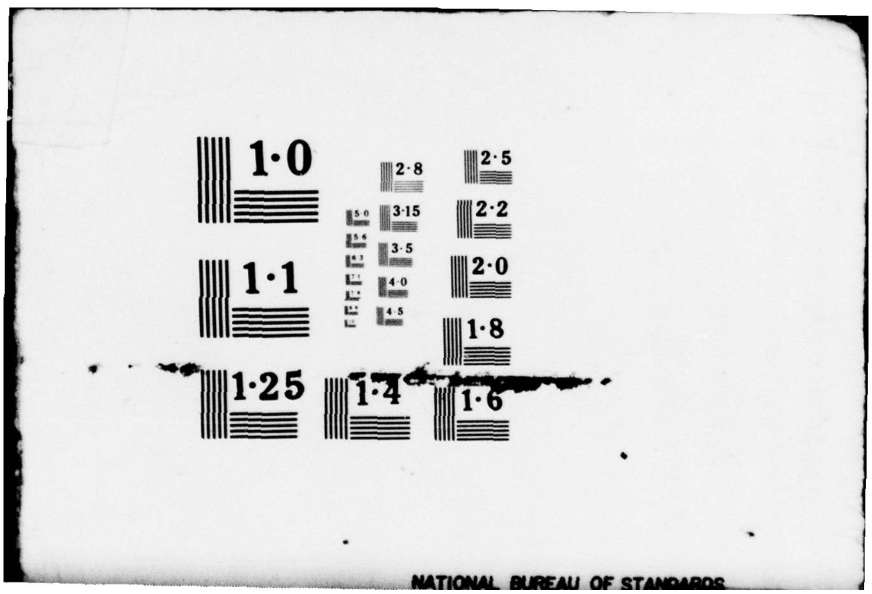
AD-A072 390 SOUTHERN METHODIST UNIV DALLAS TX DEPT OF OPERATIONS--ETC F/G 12/1
BETA VARIATE GENERATION VIA EXPONENTIAL MAJORIZING FUNCTIONS.(U)
DEC 78 B SCHMEISER, A J BABU N00014-77-C-0425
UNCLASSIFIED OREM-78014 NL

1 OF 1
AD
A072390

END
DATE
FILMED
9-79
DDC


1·0
2·8
2·5
3·15
2·2
3·5
1·1
4·0
2·0
4·5
1·8
1·25
1·4
1·6
NATIONAL BUREAU OF STANDARDS

AD A072390

LEVEL II

12

Technical Report 14 OREM-78014

6 BETA VARIATE GENERATION

VIA EXPONENTIAL MAJORIZING FUNCTIONS.

10 Bruce/Schmeiser

A. J. G./Babu

9 Technical rept.,

Department of Operations Research and Engineering Management

Southern Methodist University

Dallas, Texas 75275

12 29p.

11 Dec~~ember 1~~978

This research was supported by the Office of Naval Research,

Code 431, under contract 15 N00014-77-C-0425

ABSTRACT

Two acceptance/rejection algorithms for generating random variates from the beta distribution are developed. The algorithms use piece-wise linear and exponential majorizing functions coupled with a piece-wise linear minorizing function. The algorithms are exact to within the accuracy of the computer and are valid for all parameter values greater than one. Execution times are relatively insensitive to changes in parameter values and are faster than any previously published by roughly 50%.

Generation of beta variates having density function

$$f_\beta(x) = x^{p-1} (1-x)^{q-1} / \beta(p,q) \qquad 0 < x < 1,\ 1 < p,\ 1 < q$$

where

$$\beta(p,q) = \int_0^1 u^{p-1} (1-u)^{q-1}\, du$$

is considered here. Schmeiser and Shalaby [6] give three beta algorithms; B2P, B4P, and BNM; the latter being a modification of algorithm BN of Ahrens and Dieter [1]. The computational results in [6] indicate that in terms of execution time these three algorithms and algorithm BB of Cheng [3] are each fastest for some set of parameter values (p,q). In particular, BNM is fastest when both p and q are large and approximately equal (the distribution is approximately normal), BB is fastest when either (exclusively) p or q is large (the distribution is close to gamma), B2P is fastest when both p and q are less than two, and B4P is fastest otherwise.

In this paper algorithms B2P and B4P are modified to yield algorithms B2PE and B4PE. These algorithms dominate B2P and B4P, respectively, and both dominate BNM and BB for all parameter values. The algorithms B2PE and B4PE are developed in Section 1. Section 2 contains computational results concerning execution time and memory requirements.

1. ALGORITHM DEVELOPMENT

Algorithms B2P and B4P, as developed in [6], use piece-wise linear majorizing and minorizing functions in an acceptance/rejection algorithm. The efficiency of these algorithms is good when both p and q are relatively small, but as either or both of p and q become large the piece-wise linear majorizing function fits $f_\beta(x)$ less well and the algorithms become inefficient. The modified algorithms developed here, B2PE and B4PE, differ in the logic used to generate variates from the distribution tails. Following the logic given in Schmeiser [7], the linear majorizing function in the tail is replaced by an exponential majorizing function, as illustrated in Figures 1 and 2.

The parameters of the exponential majorizing function t(x) are chosen so that $f_\beta(x_0) = t(x_0)$ and $f_\beta'(x_0) = t'(x_0)$, where x_0 is the point at which the tail of the distribution begins. For B2PE, x_0 is the point of inflection. For B4PE, x_0 is the point at which the line tangent to $f_\beta(x)$ at the point of inflection intercepts the X axis, as shown in Figures 1 and 2. The validity of such a majorizing function follows from the Theorem, which is stated for the right tail.

Theorem. For any $x_0 \in (0,1)$, any $K > 0$, and $\lambda = q/(1-x_0) - p/x_0$,

$$K\, x_0^p\, (1-x_0)^q \exp[\lambda(x_0-x)] \geq K\, x^p\, (1-x)^q \text{ for all } x \in [x_0, 1].$$

The equality holds if and only if $x = x_0$.

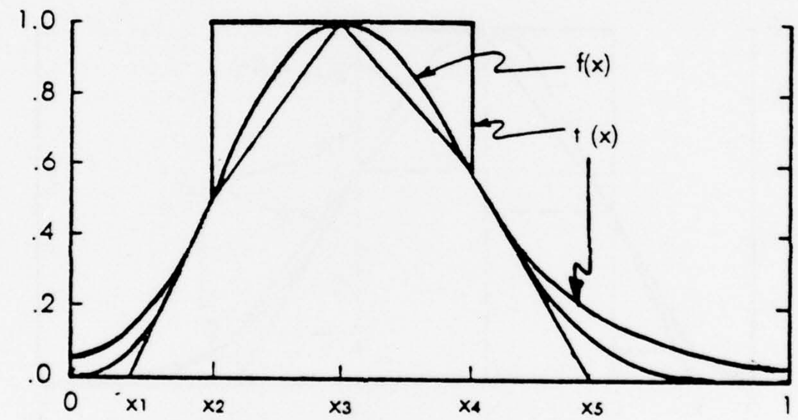


FIGURE 1. Graphical representation of algorithm B2PE.

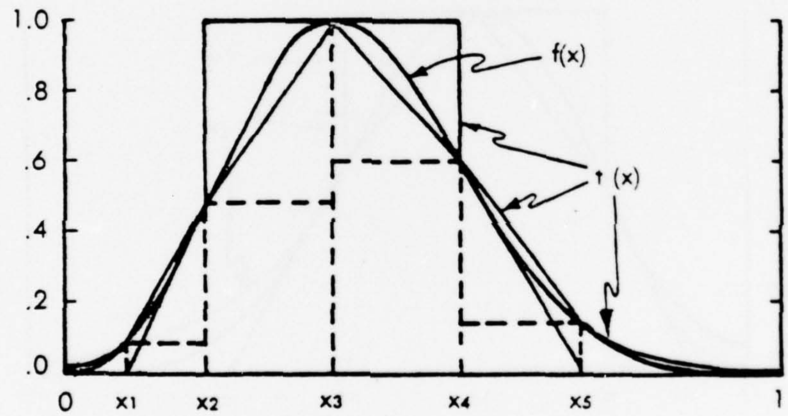


FIGURE 2. Graphical representation of algorithm B4PE.

Proof: The inequality is equivalent to

$$x_0^p (1-x_0)^q / \exp(-\lambda x_0) \geq x^p (1-x)^q / \exp(-\lambda x) \text{ for all } x \in [x_0, 1].$$

Since at $x = x_0$ the terms are equal, we need only to show that $x^p (1-x)^q \exp(\lambda x)$ is a strictly decreasing function of x. The first derivative with respect to x is

$$x^p (1-x)^q \exp(\lambda x) [p/x + \lambda - q/(1-x)]$$

which is negative if and only if

$$p/x - q/(1-x) < p/x_0 - q/(1-x_0).$$

This condition is true if and only if $p/x - q/(1-x)$ is a strictly decreasing function of x. Taking the first derivative yields

$$- [(1-x)^2 p + x^2 q] / x^2 (1-x)^2$$

which is negative, proving the theorem.

A similar theorem holds for the left tail using $\lambda = p/x_0 - q/(1-x_0)$.

Having shown the majorizing functions to be valid, we can now state the algorithms. Each algorithm is composed of two parts: set-up and generation. In the set-up the necessary constants are calculated from the parameter values p and q. The set-up is performed

only once for each set of parameter values. The generation logic is executed each time a variate is to be generated.

ALGORITHM B2PE

Set-up

1. Set $P = p-1$, $Q = q-1$, $R = P+Q$, $S = R \ln R$, $x_2 = f_2 = f_4 = 0$,
   $x_3 = P/R$, $x_4 = 1$. If $R \leq 1$, go to 4.
2. Set $D = (PQ/(R-1))^{1/2}/R$. If $D \geq x_3$, go to 3. Otherwise
   set $x_2 = x_3 - D$, $\lambda_2 = p/x_2 - q/(1-x_2)$,
   $f_2 = \exp(P \ln (x_2/P) + Q \ln ((1-x_2)/Q) + S)$.
3. If $x_3 + D \geq 1$, go to 4. Otherwise set $x_4 = x_3 + D$,
   $\lambda_4 = Q/(1-x_4) - p/x_4$, $f_4 = \exp(P \ln (x_4/P) + Q \ln ((1-x_4)/Q) + S)$.
4. Set $p_1 = x_4 - x_2$, $p_2 = f_2/\lambda_2 + p_1$, $p_3 = f_4/\lambda_4 + p_2$.

Generation

5. Sample $u \sim U(0,1)$. Set $u = up_3$. Sample $v \sim U(0,1)$.
6. If $u > p_1$, go to 7. Otherwise set $x = x_2 + u$.
   If $x < x_3$ and $v < f_2 + (x-x_2)(1-f_2)/(x_3-x_2)$, deliver x.
   If $x \geq x_3$ and $v < f_4 + (x_4-x)(1-f_4)/(x_4-x_3)$, deliver x.
   Otherwise go to 9.
7. If $u > p_2$, go to 8. Otherwise set $u = (u-p_1)/(p_2-p_1)$,
   $x = x_2 + \ln(u)/\lambda_2$. If $v < (\lambda_2(x-x_2) + 1)/u$, deliver x.
   If $x \leq 0$, go to 5. Otherwise set $v = vf_2u$ and go to 9.
8. Set $u = (u-p_2)/(p_3-p_2)$, $x = x_4 - \ln (u)/\lambda_4$.
   If $v < (\lambda_4(x_4-x) + 1)/u$, deliver x. If $x \geq 1$, go to 5.
   Otherwise set $v = vf_4u$.

9. Set $A = \ln v$. If $A > -(x-x_3)^2\ (R + R)$, go to 5.

10. If $A \le P \ln (x/P) + Q \ln ((1-x)/Q) + S$, deliver x. Otherwise go to 5.

Algorithm B4PE uses a better piece-wise fit which makes it longer and faster.

ALGORITHM B4PE

Set-up

1. Set $P = p-1$, $Q = q-1$, $R = P + Q$, $S = R \ln R$,
   $x_1 = x_2 = 0$, $x_3 = P/R$, $x_4 = x_5 = 1$,
   $f_1 = f_2 = f_4 = f_5 = 0$. If $R \le 1$, go to 4.
2. Set $D = (PQ/(R-1))^{1/2}/R$. If $D \ge x_3$, go to 3. Otherwise
   set $x_2 = x_3 - D$, $x_1 = x_2 - ((x_2(1-x_2))/(P-Rx_2))$,
   $\lambda_1 = P/x_1 - Q/(1-x_1)$,
   $f_1 = \exp(P \ln(x_1/P) + Q \ln((1-x_2)/Q) + S)$.
   $f_2 = \exp(P \ln(x_2/P) + Q \ln((1-x_2)/Q) + S)$.
3. If $x_3 + D \ge 1$, go to 4. Otherwise set
   $x_4 = x_3 + D$, $x_5 = x_4 - ((x_4(1-x_4))/(P-Rx_4))$,
   $\lambda_5 = Q/(1-x_5) - P/x_5$,
   $f_4 = \exp(P \ln(x_4/P) + Q \ln((1-x_4)/Q) + S)$,
   $f_5 = \exp(P \ln(x_5/P) + Q \ln((1-x_5)/Q) + S)$.
4. $p_1 = f_2(x_3 - x_2)$, $p_2 = f_4(x_4 - x_3) + p_1$,
   $p_3 = f_1(x_2 - x_1) + p_2$, $p_4 = f_5(x_5 - x_4) + p_3$,
   $p_5 = (1-f_2)(x_3 - x_2) + p_4$, $p_6 = (1-f_4)(x_4 - x_3) + p_5$,
   $p_7 = (f_2 - f_1)(x_2 - x_1)/2 + p_6$, $p_8 = (f_4 - f_5)(x_5 - x_4)/2 + p_7$,
   $p_9 = f_1/\lambda_1 + p_8$, $p_{10} = f_5/\lambda_5 + p_9$.

Generation

5. Sample $u \sim U(0,1)$. Set $u = up_{10}$. if $u > p_4$, go to 10.

6. If $u > p_1$, go to 7. Otherwise deliver $x = x_2 + u/f_2$.

7. If $u > p_2$, go to 8. Otherwise deliver $x = x_3 + (u-p_1)/f_4$.

8. If $u > p_3$, go to 9. Otherwise deliver $x = x_1 + (u-p_2)/f_1$.

9. Deliver $x = x_4 + (u-p_3)/f_5$.

10. Sample $w \sim U(0,1)$. If $u > p_5$, go to 11. Otherwise set $x = x_2 + w(x_3 - x_2)$. If $(u - p_4)/(p_5 - p_4) \leq w$, deliver x. Otherwise set $v = f_2 + (u - p_4)/(x_3 - x_2)$ and go to 16.

11. If $u > p_6$, go to 12. Otherwise set $x = x_3 + w(x_4 - x_3)$. If $(p_6 - u)/(p_6 - p_5) \geq w$, deliver x. Otherwise set $v = f_4 + (u - p_5) / (x_4 - x_3)$ and go to 16.

12. If $u > p_8$, go to 14.
    Otherwise sample $w_2 \sim U(0,1)$. If $w_2 > w$, set $w = w_2$.
    If $u > p_7$, go to 13. Otherwise set $x = x_1 + w(x_2 - x_1)$, $v = f_1 + 2w(u-p_6)/(x_2 - x_1)$. If $v < f_2w$, deliver x.
    Otherwise go to 16.

13. Set $x = x_5 - w(x_5 - x_4)$, $v = f_5 + 2w(u - p_7)/(x_5 - x_4)$.
    If $v \leq f_4w$, deliver x. Otherwise go to 16.

14. If $u > p_9$, go to 15. Otherwise set $u = (p_9 - u)/(p_9 - p_8)$, $x = x_1 + \ln(u)/\lambda_1$. If $x \leq 0$, go to 5.
    If $w \leq (\lambda_1(x - x_1) +1)/u$, deliver x.
    Otherwise set $v = wf_1u$ and go to 16.

15. Set $u = (p_{10} - u)/(p_{10} - p_9)$, $x = x_5 - \ln(u)/\lambda_5$.
    If $x \geq 1$, go to 5. If $w \leq (\lambda_5(x_5 - x) + 1)/u$, deliver x.
    Otherwise set $v = wf_5u$.

16. Set $A = \ln v$. If $A > -(x - x_3)^2(R + R)$, go to 5.

17. If $A \leq P \ln(x/P) + Q \ln((1-x)/Q) + S$, deliver x. Otherwise go to 5.

As can be seen by examining the algorithms, composition is used to generate the variates from the density corresponding to the majorizing function. In B2PE three regions compose the density, while in B4PE ten regions are used. Their areas are calculated in step 4 in both algorithms. Four regions in B4PE lie entirely under $f_\beta(x)$ and require no rejection logic. Thus in many cases no exponential operations are required to generate the beta variates.

Step 9 of B2PE and step 16 of B4PE also deserve comment. Here a preliminary rejection comparison is made against the normal majorizing function used by Ahrens and Dieter [1]. Thus these algorithms use both easy rejection and easy acceptance functions.

2. COMPUTATIONAL RESULTS

The two algorithms developed in Section 1 are compared to algorithms B2P, B4P, and BB in this section. Other beta variate algorithms; such as given in Atkinson and Pearce [2], Fox [4], Jöhnk [5] and Whittaker [8]; are not considered here since it was shown in [6] that the algorithms considered here dominate all others in terms of execution speed.

The computational results of this section are based on FORTRAN implementations using the FTN compiler on Southern Methodist University's CDC CYBER 72 computer. The U(0,1) psuedo-random numbers were generated by the function subprogram RANF intrinsic in the FTN compiler.

The Table summarizes the computational results. Both B2PE and B4PE dominate B2P and B4P, respectively. In addition, both B2PE and B4PE dominate BB, with B4PE being about twice as fast as BB.

The Table does not include timing information regarding set-up. For all parameter values BB requires the least set-up time. Thus the fastest algorithm is dependent upon the number of variates, M, to be generated for fixed parameter values. B4PE requires the least total time if any of the following three conditions are satisfied: 1) $M \geq 10$, 2) $M \geq 5$ and $\text{Min}(p, q) \leq 2$, or 3) $p \leq 2$ and $q \leq 2$. Otherwise BB is fastest.

In terms of memory requirements, B2P and B2PE are nearly equivalent, as are B4P and B4PE. BB requires both the least memory

TABLE I

Comparison of the Beta Generation Algorithms

(in microseconds)

Parameter Values		Schmeiser and Shalaby		Schmeiser and Babu		Cheng
p	q	B2P	B4P	B2PE	B4PE	BB
1.0001	1.0001	.37	.37	.37	.36	.56
1.01	1.01	.36	.37	.37	.37	.57
	1.5	.35	.34	.36	.43	.63
	2.	.48	.46	.43	.46	.65
	5.	.88	.42	.57	.31	.72
	10.	1.7	.72	.54	.32	.75
	100.	15.7	5.7	.51	.33	.79
1.2	1.2	.38	.39	.38	.38	.58
	1.5	.41	.41	.38	.39	.57
	2.	.41	.41	.39	.41	.58
	5.	.45	.30	.39	.28	.62
	10.	.78	.41	.43	.30	.64
	100.	6.8	2.1	.47	.31	.70
2.	2.	.43	.44	.42	.44	.61
	5.	.42	.33	.39	.30	.63
	10.	.66	.38	.42	.30	.67
	100.	4.8	1.6	.48	.33	.67
5.	5.	.33	.24	.33	.23	.62
	10.	.43	.26	.35	.24	.60
	100.	2.5	.82	.41	.27	.59
10.	10.	.42	.27	.35	.23	.65
	100.	1.8	.62	.43	.25	.59
100.	100.	1.2	.43	.35	.24	.65
1000.	1000.	3.5	.93	.36	.23	.65
Memory Requirements		498	655	467	690	335

and the least number of lines of code.

No comparison of accuracy is made since all algorithms discussed in this paper are exact to within the accuracy of the computer.

ACKNOWLEDGMENT

This research was supported by the Office of Naval Research, Code 431, under contract N00014-77-C-0425.

REFERENCES

1. J. H. Ahrens and U. Dieter, "Computer Methods for Sampling from Gamma, Beta, Poisson and Binomial Distributions," Computing, 12, 223-246 (1974).

2. A. C. Atkinson and M. C. Pearce, "The Computer Generation of Beta, Gamma and Normal Random Variables," Journal of the Royal Statistical Society, A, 139, 431-461 (1976).

3. R. C. H. Cheng, "Generating Beta Variates with Nonintegral Shape Parameters," Communications of the ACM, 21, 317-322 (1978).

4. B. L. Fox, "Generation of Random Samples from the Beta and F Distributions," Technometrics, 5, 269-270 (1963).

5. M. D. Jöhnk, "Erzeugung von Betaverteilten und Gammaverteilten Zufallszahlen," Metrika, 8, 5-15 (1964).

6. B. W. Schmeiser and M. A. Shalaby, "Acceptance/Rejection Methods for Beta Variate Generation," Technical Report 77014, Department of Operations Research and Engineering Management, Southern Methodist University, Dallas, TX 75275 (1977).

7. B. W. Schmeiser, "Generation of Variates from Distribution Tails," Technical Report 78008, Department of Operations Research and Engineering Management, Southern Methodist University, Dallas, TX 75275 (1978).

8. J. Whittaker, "Generating Gamma and Beta Random Variables with Nonintegral Shape Parameters," Applied Statistics, 23, 210-214 (1974).

COMPUTER CODES FOR

BETA VARIATE GENERATION

VIA EXPONENTIAL MAJORIZING FUNCTIONS

Bruce Schmeiser

A. J. G. Babu

Department of Operations Research and Engineering Management

Southern Methodist University

Dallas, Texas 75275

December 1978

```
      PROGRAM MAIN (INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C        THIS IS THE MAIN ROUTINE TO TEST VARIOUS METHODS OF
C        GENERATING BETA RANDOM VARIABLES
C
C        SCHMEISER AND SHALABY      JUNE 1977
C
      DIMENSION NAME(7),METHOD(7)
      DATA NAME/*BNM*,*B2P*,*B4P*,*B2PE*,*B4PE*,*BB*,* */
   10 READ(5,1000) P,Q,N,METHOD
 1000 FORMAT (2F10.3,I10,7I1)
      IF (EOF(5)) 999,20
   20 TMEAN=P/(P+Q)
      TVAR=P*Q/((P+Q)*(P+Q)*(P+Q+1))
      N = 1000
      STE=SQRT(TVAR/N)
      WRITE (6,3000) P,Q,N,TMEAN,TVAR,STE
 3000 FORMAT (*)..........................................................*,/,
     * * P=*,F6.3,*     Q=*,F8.2,*      SAMPLE SIZE=*,I5,/,
     *11X,*TIME        MEAN    VARIANCE    STD ERROR*,/,
     * * TRUE            *, 3F10.3)
C
      DO 100 I=2,6
      IF (METHOD(I) .NE. 0)  GO TO 100
      SUMT=0.
      DO 150 J=1,4
      SUM=0.
      SUM2=0.
      TIME=SECOND(X)
C
      DO 300 K=1,N
      GO TO (1,2,3,4,5,6,7), I
    1 CALL BNM(P,Q,ISEED,Y)
      GO TO 200
    2 CALL B2P(P,Q,ISEED,Y)
      GO TO 200
    3 CALL B4P(P,Q,ISEED,Y)
      GO TO 200
    4 CALL B2PE(P,Q,ISEED,Y)
      GO TO 200
    5 CALL B4PE(P,Q,ISEED,Y)
      GO TO 200
    6 CALL BB (P,Q,ISEED,Y)
      GO TO 200
    7 CALL BB (P,Q,ISEED,Y)
  200 SUM=SUM+Y
  300 SUM2=SUM2+Y*Y
C
      TIME=1000.*(SECOND(X)-TIME)/N
      SUMT=SUMT+TIME
      AVGT=SUMT/J
      XMEAN=SUM/N
      VAR=SUM2/N-XMEAN*XMEAN
  150 WRITE (6,2000)  NAME(I),TIME,AVGT,XMEAN,VAR
 2000 FORMAT(* *,A4,2F6.2,F8.3,F10.3)
  100 CONTINUE
      GO TO 10
C
  999 STOP
      END
```

```
      SUBROUTINE B2P (P,Q,ISEED,X)
C
C        GENERATION OF ONE PSEUDO-RANDOM VARIATE USING
C        THE TWO POINT TECHNIQUE DESCRIBED IN SECTION 2.2
C        FROM THE BETA DENSITY FUNCTION PROPORTIONAL TO
C           F(X) = (X**(P-1)) * ((1-X)**(Q-1))
C
C        P = FIRST PARAMETER    (GREATER THAN ONE)
C        Q = SECOND PARAMETER   (GREATER THAN ONE)
C        ISEED = A RANDOM NUMBER SEED
C        X = THE BETA VARIATE
C
      DATA PSAVE, QSAVE /-1.,-1./
C
C        CHECK WHETHER SET-UP IS NECESSARY
C
      IF (P .EQ. PSAVE .AND. Q .EQ. QSAVE)  GO TO 100
C
C        PERFORM SET-UP
C
      PSAVE=P
      QSAVE=Q
      PP=P-1.
      QQ=Q-1.
      RR=PP+QQ
      TRR=-R2-RR
      RL=RR*ALOG(RR)
      Y1=0.
      Y2=0.
      Y3=PP/RR
      Y4=1.
      Y5=1.
      F2=0.
      F4=0.
C
      IF (RR .LE. 1.)  GO TO 80
      D=SQRT(PP*QQ/(RR-1.))/RR
      IF(D .GE. Y3)   GO TO 60
      Y2=Y3-D
      Y1=Y2-((Y2*(1.-Y2))/(PP-RR*Y2))
      F2=EXP(PP*ALOG(Y2/PP)+QQ*ALOG((1.-X2)/QQ)+RL)
   60 IF (D .GE. (1.-Y3))   GO TO 80
      Y4=Y3+D
      Y5=Y4-((Y4*(1.-Y4))/(PP-RR*Y4))
      F4=EXP(PP*ALOG(Y4/PP)+QQ*ALOG((1.-X4)/QQ)+RL)
C
C        CALCULATE AREAS OF THE FOUR REGIONS
C
   80 R1=Y3-Y2
      R2=(Y4-Y3)+R1
      R3=R2
      IF(F2 .GT. 0.)  R3=F2*Y2*.5+R2
      R4=R3
      IF(F4 .GT. 0.)  R4=F4*(1.-Y4)*.5+R3
```

```
C
C         REJECTION PROCEDURE BEGINS HERE
C
  100 U=RAND(ISEED) * P4
      W=RAND(ISEED)
C
C         LEFT RECTANGULAR REGION
C
      IF(U.GT.P1)GO TO 200
      X=X2+W*(X3-X2)
      V=U/P1
      IF(V .LE. F2+(1.-F2)*W)  RETURN
      GO TO 500
C
C         RIGHT RECTANGULAR REGION
C
  200 IF(U.GT.P2)GO TO 300
      X=X3+W*(X4-X3)
      V=(U-P1)/(P2-P1)
      IF (V .LE. 1.-(1.-F4)*W)  RETURN
      GO TO 500
C
C         LEFT TRIANGULAR REGION
C
  300 W2=RAND(ISEED)
      IF(W2.GT.W)W=W2
      IF(U.GT.P3) GO TO 400
      X=W*X2
      V=((U-P2)/(P3-P2))*W*F2
      IF (X .LE. X1)  GO TO 500
      IF(V.LE.F2*(X-X1)/(X2-X1))RETURN
      GO TO 500
C
C         RIGHT TRIANGULAR REGION
C
  400 X=1.-W*(1.-X4)
      V=((U-P3)/(P4-P3))*(1.-X)*F4/(1.-X4)
      IF (X .GT. X5)  GO TO 500
      IF(V.LE.F4*(X5-X)/(X5-X4))RETURN
C
C         CHECK EASY REJECTION VIA COMPARISON WITH THE NORMAL DENSITY
C
  500 ALV=ALOG(V)
      IF (ALV .GT. (X-X3)*(X-X3)*TPP)  GO TO 100
C
C         PERFORM REJECTION BASED ON F(X)
C
      IF(ALV.LE.PP*ALOG(X/PP)+QQ*ALOG((1.-X)/QQ)+RL)  RETURN
      GO TO 100
      END
```

```
      SUBROUTINE B4P (P,Q,ISEED,X)
C
C        GENERATION OF ONE PSEUDO-RANDOM VARIATE USING
C        THE FOUR POINT METHOD DESCRIBED IN SECTION 2.3
C        FROM THE BETA DENSITY FUNCTION PROPORTIONAL TO
C           F(X) = (X**(P-1)) * ((1-X)**(Q-1))
C
C        P = FIRST PARAMETER    (GREATER THAN ONE)
C        Q = SECOND PARAMETER   (GREATER THAN ONE)
C        ISEED = A RANDOM NUMBER SEED
C        X = THE BETA VARIATE
C
      DATA PSAVE, QSAVE /-1.,-1./
C
C        CHECK WHETHER SET-UP IS NECESSARY
C
      IF (P .EQ. PSAVE .AND. Q .EQ. QSAVE)  GO TO 100
C
C        PERFORM SET-UP
C
      PSAVE=P
      QSAVE=Q
      PP=P-1.
      QQ=Q-1.
      RR=PP+QQ
      TRR=-RR-RR
      RL=RR*ALOG(RR)
      X1=0.
      X2=0.
      X3=PP/RR
      X4=1.
      X5=1.
      F1=0.
      F2=0.
      F4=0.
      F5=0.
C
      IF (RR .LE. 1.)  GO TO 80
      D=SQRT(PP*QQ/(RR-1.))/RR
      IF(D .GE. X3)  GO TO 60
      X2=X3-D
      X1=X2-((X2*(1.-X2))/(PP-RR*X2))
      F1=EXP(PP*ALOG(X1/PP)+QQ*ALOG((1.-X1)/QQ)+RL)
      F2=EXP(PP*ALOG(X2/PP)+QQ*ALOG((1.-X2)/QQ)+RL)
   60 IF(D .GE. (1.-X3))  GO TO 80
      X4=X3+D
      X5=X4-((X4*(1.-X4))/(PP-RR*X4))
      F4=EXP(PP*ALOG(X4/PP)+QQ*ALOG((1.-X4)/QQ)+RL)
      F5=EXP(PP*ALOG(X5/PP)+QQ*ALOG((1.-X5)/QQ)+RL)
```

```
C
C        CALCULATE AREAS OF THE TEN REGIONS
C
 80 P1=0.
    IF(F2 .GT. 0.)  P1=F2*(X3-X2)
    P2=P1
    IF(F4 .GT. 0.)  P2=F4*(X4-X3)+P1
    P3=P2
    IF(F1 .GT. 0.)  P3=F1*(X2-X1)+P2
    P4=P3
    IF(F5 .GT. 0.)  P4=F5*(X5-X4)+P3
    P5=(1.-F2)*(X3-X2)+P4
    P6=(1.-F4)*(X4-X3)+P5
    P7=P6
    IF(F2 .GT. F1)  P7=(F2-F1)*(X2-X1)*.5+P6
    P8=P7
    IF(F4 .GT. F5)  P8=(F4-F5)*(X5-X4)*.5+P7
    P9=P8
    IF(F1 .GT. 0.)  P9=F1*X1*.5+P8
    P10=P9
    IF(F5 .GT. 0.)  P10=F5*(1.-X5)*.5+P9
C
C        REJECTION PROCEDURE BEGINS HERE
C
100 U=RANF(ISEED) * P10
C
C        THE FOUR REGIONS WITH ZERO PROBABILITY OF REJECTION
C
    IF(U.GT.P4)GO TO 500
    IF(U.GT.P1)GO TO 200
    X=X2+U/F2
    RETURN
200 IF(U.GT.P2)GO TO 300
    X=X3+(U-P1)/F4
    RETURN
300 IF(U.GT.P3)GO TO 400
    X=X1+(U-P2)/F1
    RETURN
400 X=X4+(U-P3)/F5
    RETURN
```

```
C
C          THE TWO REGIONS USING RECTANGULAR REJECTION
C
  500 W=RANF(ISEED)
      IF(U.GT.P5)GO TO 600
      X=X2+(X3-X2)*W
      IF((U-P4)/(P5-P4) .LE. W)  RETURN
      V=F2+(U-P4)/(X3-X2)
      GO TO 1300
  600 IF(U.GT.P6)GO TO 700
      X=X3+(X4-X3)*W
      IF((P6-U)/(P6-P5) .GE. W)  RETURN
      V=F4+(U-P5)/(X4-X3)
      GO TO 1300
C
C          THE FOUR TRIANGULAR REGIONS
C
  700 W2=RANF(ISEED)
      IF(W2.GT.W)W=W2
      IF(U.GT.P7)GO TO 800
      X=X1+(X2-X1)*W
      V=F1+2.*W*(U-P6)/(X2-X1)
      IF (V .LE. F2*W)  RETURN
      GO TO 1300
  800 IF(U.GT.P8)GO TO 900
      X=X5-W*(X5-X4)
      V=F5+2.*W*(U-P7)/(X5-X4)
      IF (V .LE. F4*W)  RETURN
      GO TO 1300
  900 IF(U.GT.P9)GO TO 1000
      X=W*X1
      V=2.*W*(U-P8)/X1
      GO TO 1300
 1000 X=1.-W*(1.-X5)
      V=2.*W*(U-P9)/(1.-X5)
C
C          CHECK EASY REJECTION VIA COMPARISON WITH NORMAL DENSITY FUNCTION
C
 1300 ALV=ALOG(V)
      IF (ALV .GT. (X-X3)*(X-X3)*TRP)  GO TO 100
C
C          PERFORM THE STANDARD REJECTION
C
      IF(ALV.LE.PP*ALOG(X/PP)+QQ*ALOG((1.-X)/QQ)+RL)RETURN
      GO TO 100
      END
```

```
      SUBROUTINE B2PE(P,Q,ISEED,X)
C ***
C *** B. SCHMEISER AND A.J.G. BABU     JULY 1978    SMU
C *** GENERATION OF A PSEUDO-RANDOM VARIATE
C *** USING THE TWO POINT METHOD WITH EXPONENTIAL TAILS
C *** WITH BETA DENSITY FUNCTION PROPORTIONAL TO
C ***
C ***        F(Y) = (Y**(P-1)) * ((1-Y)**(Q-1))
C ***
C *** P = FIRST PARAMETER    (GRATER THAN ONE)
C *** Q = SECOND PARAMETER   (GRATER THAN ONE)
C *** ISEED = A RANDOM NUMBER SEED
C *** X = THE BETA VARIATE
C ***
      DATA PSAVE,QSAVE,A2,A4/-1.,-1.,1.,1./
C ***
C *** CHECK WHETHER SET UP IS NECESSARY
C ***
      IF(P.EQ.PSAVE.AND.Q.EQ.QSAVE) GO TO 100
C ***
C *** PERFORM SET UP
C ***
      PSAVE=P
      QSAVE=Q
      PP=P-1
      QQ=Q-1
      RR=PP+QQ
      TRR=-RR-RR
      RL=RR*ALOG(RR)
      X2 = F2 = F4 = 0
      X3=PP/RR
      X4 = 1
C ***
      IF(RR.LE.1) GO TO 80
      D=SQRT(PP*QQ/(RR-1))/RR
      IF(D.GE.X3) GO TO 60
      X2=X3-D
      A2=(PP/X2)-(QQ/(1-X2))
      F2=EXP(PP*ALOG(X2/PP)+QQ*ALOG((1-X2)/QQ)+RL)
   60 IF(D.GE.(1-X3)) GO TO 80
      X4=X3+D
      A4=(QQ/(1-X4))-(PP/X4)
      F4=EXP(PP*ALOG(X4/PP)+QQ*ALOG((1-X4)/QQ)+RL)
C ***
C *** CALCULATE AREAS OF THE THREE REGIONS
C ***
   80 P1=X4-X2
      P2=F2/A2+P1
      P3=F4/A4+P2
```

```
C ***
C *** REJECTION PROCEDURE BEGINS HERE
C ***
  100 U=RANF(ISEED)*P3
      V=RANF(ISEED)
C ***
C *** RECTANGULAR REGION
C ***
      IF(U.GT.P1) GO TO 200
      X=X2+U
      IF (X .GT. X3)  GO TO 150
      IF(V.LT.(F2+(X-X2)*(1-F2)/(X3-X2)))RETURN
      GO TO 400
  150 IF(V.LT.(F4+(X4-X)*(1-F4)/(X4-X3)))RETURN
      GO TO 400
C ***
C *** LEFT EXPONENTIAL TAIL
C ***
  200 IF(U.GT.P2) GO TO 300
      U=(U-P1)/(P2-P1)
      X=X2+ALOG(U)/A2
      IF (V .LT. (A2*(X-X2)+1)/U)  RETURN
      IF(X.LE.0) GO TO 100
      V=V*F2*U
      GO TO 400
C ***
C *** RIGHT EXPONENTIAL TAIL
C ***
  300 U=(U-P2)/(P3-P2)
      X=X4-ALOG(U)/A4
      IF (V .LT. (A4*(X4-X)+1)/U)  RETURN
      IF(X.GE.1.) GO TO 100
      V=V*F4*U
C ***
C *** TEST V VS. NORMAL AND BETA DENSITIES
C ***
  400 ALV=ALOG(V)
      IF(ALV.GT.((X-X2)*(X-X2)*TPP)) GO TO 100
      IF(ALV.LT.(PP*ALOG(X/PP)+QQ*ALOG((1-X)/QQ)+PL))RETURN
      GO TO 100
      END
```

```
      SUBROUTINE B4PE(P,Q,ISEED,X)
C
C        GENERATION OF ONE PSEUDO-RANDOM VARIATE USING
C        THE FOUR POINT METHOD WITH EXPONENTIAL TAILS
C        FROM THE BETA DENSITY FUNCTION PROPORTIONAL TO
C           F(X) = (X**(P-1)) * ((1-X)**(Q-1))
C
C        P = FIRST PARAMETER    (GREATER THAN ONE)
C        Q = SECOND PARAMETER   (GREATER THAN ONE)
C        ISEED = A RANDOM NUMBER SEED
C        X = THE BETA VARIATE
C
C        REFERENCE : BRUCE W. SCHMEISER AND A.J.G.BABU
C
      DATA PSAVE,QSAVE,A1,A5/-1.,-1.,1.,1./
C
C        CHECK WHETHER SET-UP IS NECESSARY
C
      IF (P .EQ. PSAVE .AND. Q .EQ. QSAVE)  GO TO 100
C
C        PERFORM SET-UP
C
      PSAVE=P
      QSAVE=Q
      PP=P-1.
      QQ=Q-1.
      RR=PP+QQ
      TRR=-RR-RR
      C=RR*ALOG(RR)-PP*ALOG(PP)-QQ*ALOG(QQ)
      X1=X2=F1=F2=F4=F5=0.
      X3=PP/RR
      X4=X5=1.
C
      IF (RR .LE. 1.)  GO TO 80
      D=SQRT(PP*QQ/(RR-1.))/RR
      IF(D .GE. X3)  GO TO 60
      X2=X3-D
      X1=X2-(X2*(1.-X2))/(PP-RR*X2)
      A1=PP/X1-QQ/(1-X1)
      F1=EXP(C+PP*ALOG(X1)+QQ*ALOG(1-X1))
      F2=EXP(C+PP*ALOG(X2)+QQ*ALOG(1-X2))
   60 IF(D .GE. (1.-X3))  GO TO 80
      X4=X3+D
      X5=X4-((X4*(1.-X4))/(PP-RR*X4))
      A5=QQ/(1-X5)-PP/X5
      F4=EXP(C+PP*ALOG(X4)+QQ*ALOG(1-X4))
      F5=EXP(C+PP*ALOG(X5)+QQ*ALOG(1-X5))
```

```
C
C        CALCULATE AREAS OF THE TEN REGIONS
C
  80 P1=0.
     IF(F2 .GT. 0.)  P1=F2*(X3-X2)
     P2=P1
     IF(F4 .GT. 0.)  P2=F4*(X4-X3)+P1
     P3=P2
     IF(F1 .GT. 0.)  P3=F1*(X2-X1)+P2
     P4=P3
     IF(F5 .GT. 0.)  P4=F5*(X5-X4)+P3
     P5=(1.-F2)*(X3-X2)+P4
     P6=(1.-F4)*(X4-X3)+P5
     P7=P6
     IF(F2 .GT. F1)  P7=(F2-F1)*(X2-X1)*.5+P6
     P8=P7
     IF(F4 .GT. F5)  P8=(F4-F5)*(X5-X4)*.5+P7
     P9=P8
     IF(F1.GT.0) P9=F1/A1+P8
     P10=P9
     IF(F5.GT.0) P10=F5/A5+P9
C
C        REJECTION PROCEDURE BEGINS HERE
C
 100 U=RANF(ISEED) * P10
C
C        THE FOUR REGIONS WITH ZERO PROBABILITY OF REJECTION
C
     IF(U.GT.P4)GO TO 500
     IF(U.GT.P1)GO TO 200
     X=X2+U/F2
     RETURN
 200 IF(U.GT.P2)GO TO 300
     X=X3+(U-P1)/F4
     RETURN
 300 IF(U.GT.P3)GO TO 400
     X=X1+(U-P2)/F1
     RETURN
 400 X=X4+(U-P3)/F5
     RETURN
C
C        THE TWO REGIONS USING RECTANGULAR REJECTION
C
 500 W=RANF(ISEED)
     IF(U.GT.P5)GO TO 600
     X=X2+(X3-X2)*W
     IF((U-P4)/(P5-P4) .LE. W)  RETURN
     V=F2+(U-P4)/(X3-X2)
     GO TO 1300
 600 IF(U.GT.P6)GO TO 700
     X=X3+(X4-X3)*W
     IF((P6-U)/(P6-P5) .GE. W)  RETURN
     V=F4+(U-P5)/(X4-X3)
     GO TO 1300
```

```
C
C          THE TWO TRIANGULAR REGIONS
C
  700 IF(U.GT.P8) GO TO 900
      W2=RAN(ISEED)
      IF(W2.GT.W)W=W2
      IF(U.GT.P7)GO TO 800
      X=X1+(X2-X1)*W
      V=F1+2.*W*(U-P6)/(X2-X1)
      IF (V .LE. F2*W)  RETURN
      GO TO 1300
  800 X=X5-W*(X5-X4)
      V=F5+2.*W*(U-P7)/(X5-X4)
      IF (V .LE. F4*W)  RETURN
      GO TO 1300
C
C          THE TWO EXPONENTIAL REGIONS
C
  900 IF(U.GT.P9)GO TO 1000
      U=(P9-U)/(P9-P8)
      X=X1+ALOG(U)/A1
      IF (W .LT. (A1*(X-X1)+1.)/U)  RETURN
      IF(X.LE.0.) GO TO 100
      V=W*F1*U
      GO TO 1300
 1000 U=(P10-U)/(P10-P9)
      X=X5-ALOG(U)/A5
      IF (W .LT. (A5*(X5-X)+1.)/U)  RETURN
      IF(X.GE.1.) GO TO 100
      V=W*F5*U
C
C          CHECK EASY REJECTION VIA COMPARISON WITH NORMAL DENSITY FUNCTION
C
 1300 ALV=ALOG(V)
      IF (ALV .GT. (X-X3)*(X-X3)*TPP)  GO TO 100
C
C          PERFORM THE STANDARD REJECTION
C
      IF(ALV.LE.C+PP*ALOG(X)+QQ*ALOG(1.-X))RETURN
      GO TO 100
      END
```

```
      SUBROUTINE BR (P,Q,ISEED,X)
C
C        GENERATION OF ONE PSEUDO-RANDOM VARIATE USING
C        THE METHOD OF CHENG, CACM, APRIL 1978, 317-322.
C        FROM THE BETA DENSITY PROPORTIONAL TO
C           F(X) = (X**(P-1)) * ((1-X)**(Q-1))
C
C        P = FIRST PARAMETER
C        Q = SECOND PARAMETER
C        ISEED = A RANDOM NUMBER SEED
C        X = THE BETA VARIATE
C
      DATA PSAVE, QSAVE /-1.,-1./
C
C        CHECK WHETHER SET-UP IS NECESSARY
C
      IF (P .EQ. PSAVE .AND. Q .EQ. QSAVE)  GO TO 100
C
C        PERFORM SET-UP
C
      PSAVE = P
      QSAVE = Q
      A = P
      B = Q
      IF ( A .LT. B)  GO TO 200
      A = Q
      B = P
  200 ALPHA = A + B
      BETA = SQRT ((ALPHA-2.) / (2.*A*B-ALPHA))
      GAMMA = A + 1./BETA
C
C        REJECTION PROCEDURE BEGINS HERE
C
  100 U1 = RANF(ISEED)
      U2 = RANF(ISEED)
      V = BETA * ALOG(U1/(1.-U1))
      W = A * EXP(V)
      Z = U1*U1*U2
      R = GAMMA * V - 1.386294361
      S = A + R - W
      IF (S + 2.609437912 .GE. 5.*Z)  GO TO 500
      T = ALOG(Z)
      IF (S .GE. T)  GO TO 500
      IF (R + ALPHA*ALOG(ALPHA/(B+W)) .LT. T)  GO TO 100
  500 IF (A .EQ. P)  X = W / (B + W)
      IF (A .NE. P)  X = B / (B + W)
      RETURN
      END
```

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: OREM 78014	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Beta Variate Generation via Exponential Majorizing Functions		5. TYPE OF REPORT & PERIOD COVERED: Technical Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Bruce W. Schmeiser; A. J. G. Babu		8. CONTRACT OR GRANT NUMBER(s): N00014-77-C-0425
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Department of Operations Research and Engineering Management, Southern Methodist University, Dallas, TX		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: NR 277-236
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Analysis Program, Code 431, Office of Naval Research, Arlington, VA 27217		12. REPORT DATE: December, 1978
		13. NUMBER OF PAGES: 28
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report): Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Process Generation
Simulation
Distribution Sampling
Monte Carlo
Random Numbers

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

Two acceptance/rejection algorithms for generating random variates from the beta distribution are developed. The algorithms use piecewise linear and exponential majorizing functions coupled with a piecewise linear minorizing function. The algorithms are exact to within the accuracy of the computer and are valid for all parameter values greater than one. Execution times are relatively insensitive to changes in parameter values and are faster than any previously published by roughly 50%.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)